Católica de Santa Catarina
Centro Universitário

RELATÓRIO FINAL

DESENVOLVIMENTO DE DISPOSITIVOS MÓVEIS

Curso: Engenharia de Software – Fase: 5ª

App de Chamada Automática

BIANCA BARP

MARINA ROSA OLIVEIRA

NATHALIA ALINE BERRI

RELATÓRIO do Projeto de Criação de um Aplicativo Mobile de Chamada Automatizada apresentado como requisito parcial de avaliação na disciplina Desenvolvimento de Dispositivos Móveis da fase 4 no curso de graduação de Engenharia de Software do Centro Universitário Católica de Santa Catarina - Campus Joinville sob supervisão e orientação do professor Diego Sauter Possamai.

JOINVILLE, DEZEMBRO DE 2024

SUMÁRIO

1 APRESENTAÇÃO - INTRODUÇÃO

O trabalho tinha como objetivo projetar e modelar um aplicativo móvel que automatize a chamada sem intervenção do professor, entregando um protótipo funcional com telas navegáveis e lógica básica. Deveriam haver 4 chamadas por noite, com janelas de aproximadamente 50min cada e o mapeamento de um método de impeça que os alunos burlem a chamada, ou seja, eles devem estar obrigatoriamente em sala. Além disso, deve ser gerado um relatório CSV por estudante/dia/rodada.

1.2 VISÃO E ESCOPO

1.2.1 VISÃO

O aplicativo Auto-Chamada tem como propósito automatizar o processo de chamada de presença em salas de aula, eliminando a necessidade de intervenção do professor. Ele busca garantir maior confiabilidade, praticidade e integridade nas presenças, por meio de chamadas automáticas em horários pré-definidos e validação de localização geográfica dos alunos.

1.2.2 ESCOPO

O sistema contempla:

- Aplicativo mobile (Android) para alunos e professores;
- Login e cadastro de usuários com autenticação segura;
- Quatro chamadas automáticas por noite, com temporizador e notificações;
- Validação de presença via geolocalização (GPS);
- Geração de relatórios CSV de presenças (por aluno e por dia);
- Interface simples e responsiva, com timers e cores indicando o status de cada chamada.

Fora do escopo:

- Integração com o Portal do Aluno da Católica.
- Controle manual de presenças por parte do professor.
- Recuperação de presenças fora da data da chamada.

1.3 STAKEHOLDERS

Stakeholder	Descrição / Papel no Sistema
Aluno	Usuário principal do app. Realiza login, responde às chamadas automáticas e pode exportar seu relatório de presenças.
Professor	Responsável por acompanhar as presenças e extrair relatórios gerais. Não precisa realizar chamadas manualmente.
Instituição (Católica SC)	Beneficiária do sistema, que ganha em eficiência e confiabilidade no controle de presenças.
Equipe de Desenvolvimento	Responsável por projetar, codificar e testar o aplicativo e sua API.
Coordenador de Curso / Secretaria Acadêmica	Pode utilizar os relatórios para conferência administrativa de frequência.

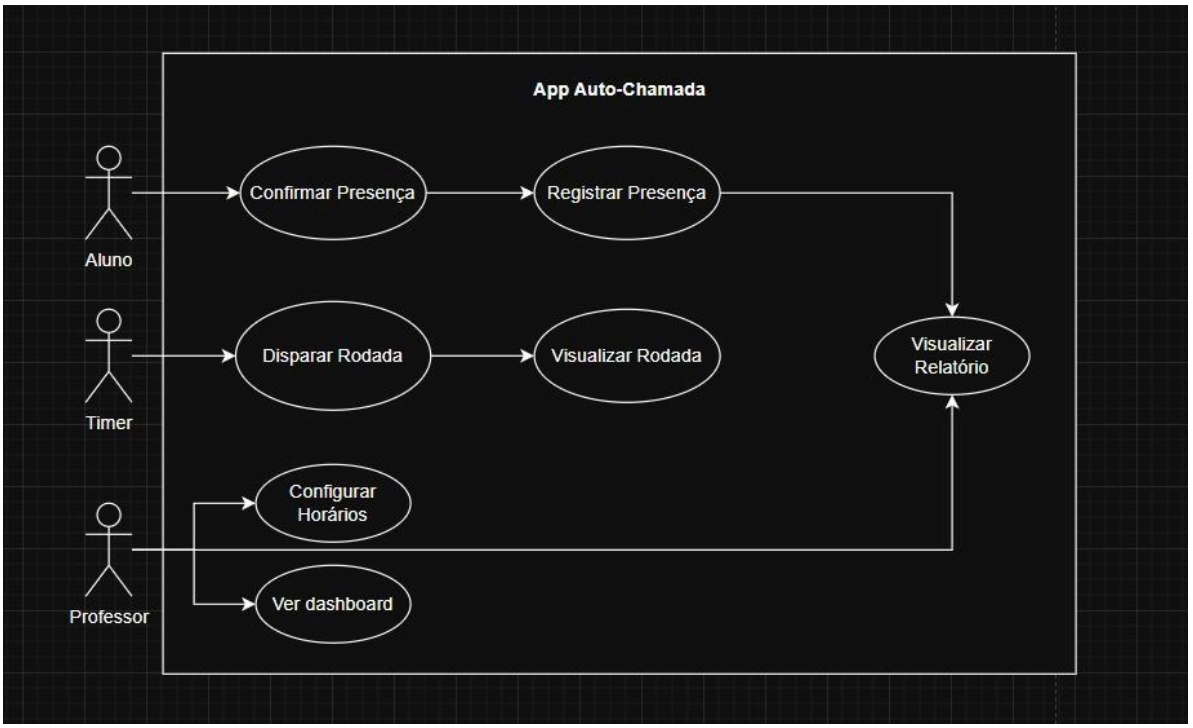
2 DIAGRAMAS E WIREFRAMES

2.1 WIREFRAMES

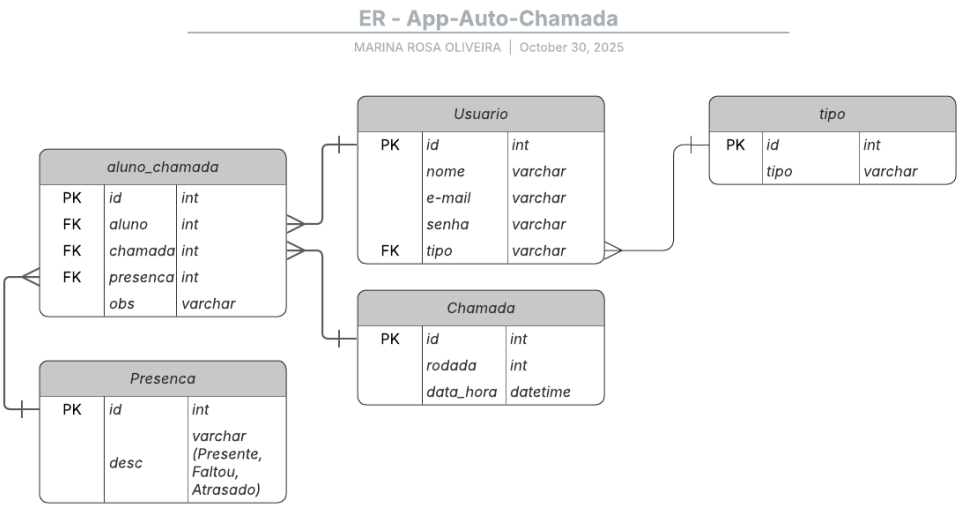
Os wireframes do projeto, assim como o fluxo de telas, podem ser encontrados em https://www.figma.com/design/0YYpOJv3WEz1IcEAteV2k3/N2---Desenvolvimento-Mobile?node-id=0-1&p=f&t=FaYsGuCZitnqn70o-0.

2.2 DIAGRAMAS

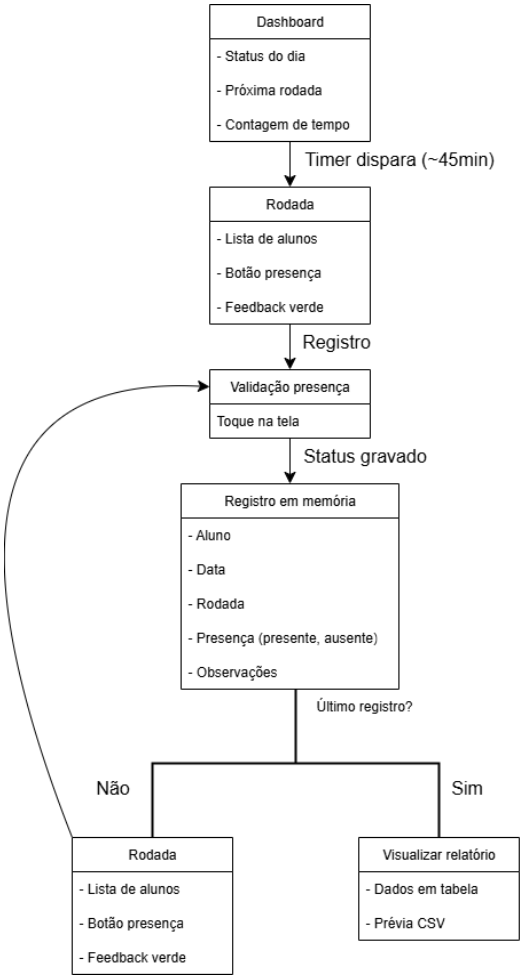
2.2.1 CASOS DE USO



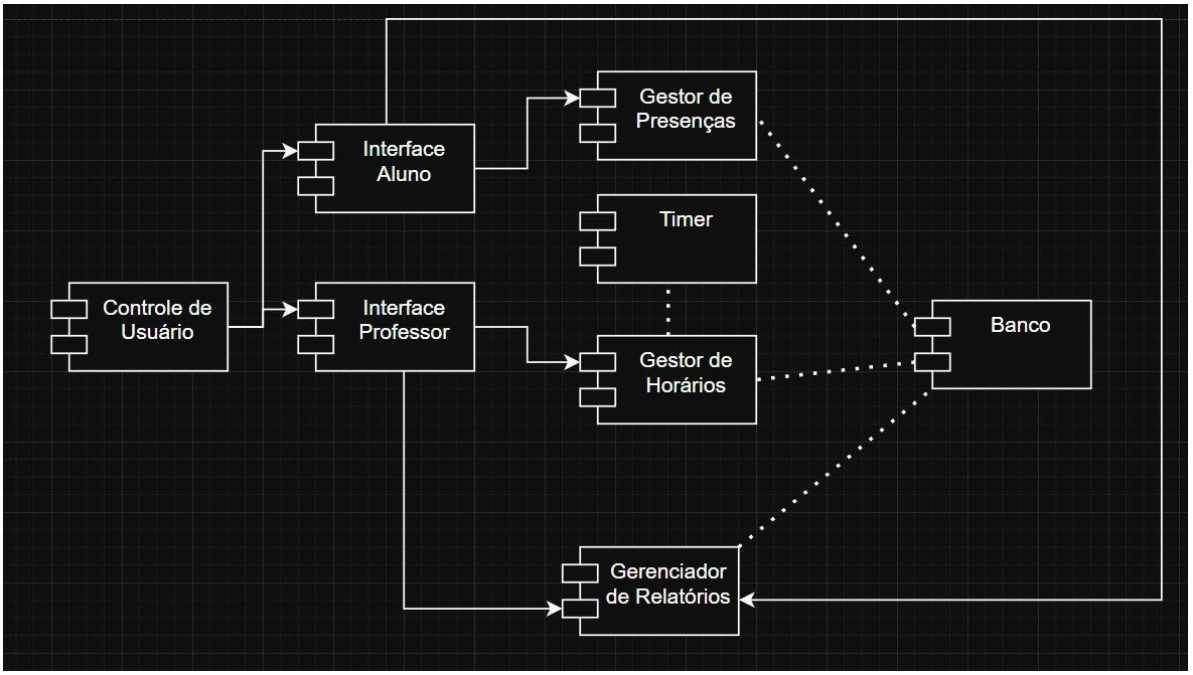
2.2.2 DIAGRAMA DE ENTIDADE-RELACIONAMENTO



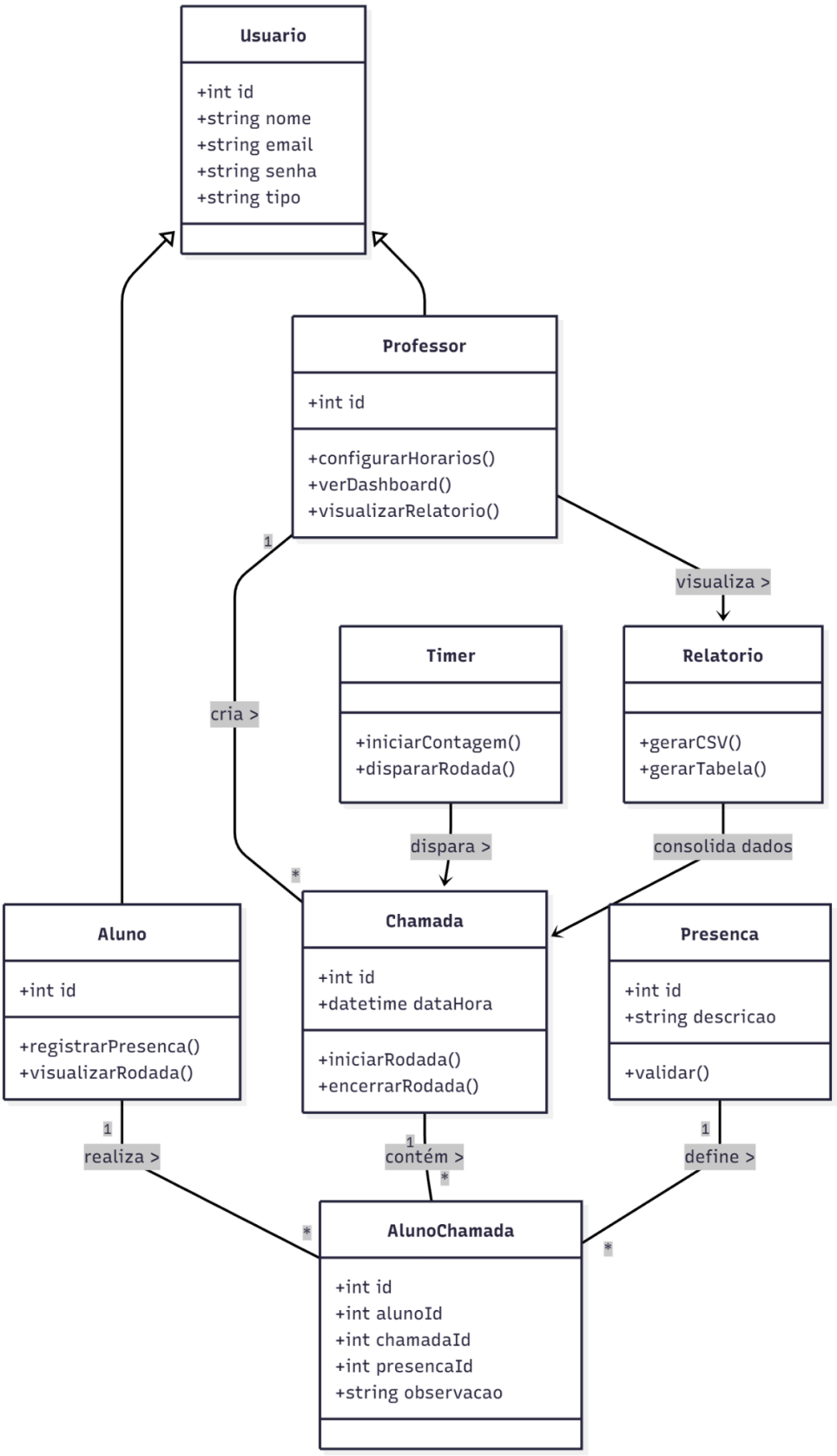
2.2.3 DIAGRAMA DE ATIVIDADES



2.2.4 DIAGRAMA DE COMPONENTES



2.2.5 DIAGRAMA DE CLASSES



3. REQUISITOS

3.1 REQUISITOS FUNCIONAIS

RF001	A tela inicial padrão deve ser a tela de Login
RF002	O usuário deve fazer cadastro antes de acessar o aplicativo, informando seu nome, e-mail e senha
RF003	Para fazer login, o usuário deve informar seu e-mail e senha corretamente
RF004	Deve haver um link direto entre as telas de Login e Cadastro
RF005	O app deve possuir uma appbar superior vermelha com a logo da católica e o nome do aplicativo (Auto-Chamada)
RF006	O app deve possuir uma appbar inferior de navegação que fica fixa na tela (não desaparece de acordo com a rolagem) com um botão que redireciona à home, um botão que permite que o usuário faça logout e um botão que redireciona-o à tela de extração de relatórios
RF007	O app deve possuir uma Home onde é mostrado o timer com uma contagem regressiva em HH:MM:SS até a próxima chamada e o horário em que cada uma será realizada
RF008	O app deve prever 4 chamadas por noite, com intervalos de 45 minutos entre cada uma, sendo a primeira às 19h45
RF009	Quando uma chamada estiver inativa, o card com seu horário deve ser cinza e não-clicável
RF010	Quando uma chamada estiver ativa, o card com seu horário deve ser azul e clicável
RF011	Quando o aluno já tiver respondido uma chamada dentro do horário limite, o card deve ficar verde e ele deve receber presença sem observação
RF012	Quando o aluno já tiver respondido uma chamada, mas fora do horário limite, o card deve ficar amarelo
RF013	Quando o aluno não tiver respondido uma chamada e o horário limite já tiver sido finalizado, o card deve ficar vermelho
RF014	O período para responder a chamada deve ser 5 minutos
RF015	Deve haver uma tolerância de mais 5 minutos para que o usuário possa

	responder a chamada atrasado, mas ainda receber presença
RF016	Após o tempo limite e tempo de tolerância de uma chamada, seu botão deve voltar a ser não-clicável
RF017	O professor deve conseguir extrair um relatório .CSV de presenças por aluno ou dia
RF018	O aluno deve conseguir extrair um relatório .CSV com seu histórico de presenças
RF019	As chamadas devem ocorrer de forma automática, sem interferência do professor
RF020	O aplicativo deve capturar a localização do aluno no momento em que ele responde a chamada para garantir que ele esteja no Campus, evitando fraudes

3.2 REQUISITOS NÃO-FUNCIONAIS

Código	Descrição
RNF001	O aplicativo deve ser desenvolvido utilizando Flutter.
RNF002	O backend deve ser desenvolvido utilizando Node.js e Express.
RNF003	O banco de dados deve ser MySQL.
RNF004	A transição entre telas deve levar menos de 3 segundos.
RNF005	O app deve funcionar em dispositivos Android a partir da versão 8.0 (Oreo).
RNF006	O sistema deve armazenar dados de forma segura, criptografando senhas com bcrypt.
RNF007	A API deve responder em tempo médio inferior a 2 segundos para requisições de chamada e login.
RNF008	O sistema deve ser capaz de operar offline temporariamente e sincronizar dados quando houver conexão.
RNF009	A localização do aluno deve ter precisão mínima de 50 metros para validação de presença.
RNF010	A interface deve seguir o Material Design e manter o padrão de cores da instituição (predominantemente vermelho e branco).

3.3 MISUSE CASES

ID	Título do Misuse Case	Descrição do Abuso Potencial	Mitigação / Contramedida
MU001	Aluno tenta marcar presença fora do campus	O aluno ativa o botão de presença sem estar fisicamente na sala.	Implementar verificação de geolocalização obrigatória e bloquear respostas fora da área delimitada (raio ≤ 50m do campus).
MU002	Aluno compartilha login com outro colega	Um aluno fornece suas credenciais para outro responder por ele.	Utilizar token de sessão exclusivo por dispositivo e expiração automática ao detectar acesso simultâneo.
MU003	Manipulação de horário do dispositivo	O aluno altera o relógio do celular para tentar responder fora do horário limite.	Validar o horário da chamada com base no servidor backend, não no dispositivo.
MU004	Tentativa de interceptar requisições HTTP	Usuário tenta enviar requisições falsas de presença via ferramentas como Postman.	Implementar autenticação JWT, HTTPS e validação de assinatura digital do token.

3.4 CRITÉRIOS DE ACEITE

ID	Critério de Aceite	Métrica / Validação
CA001	O login deve autenticar corretamente o usuário.	Teste manual com 10 tentativas válidas e 10 inválidas → taxa de sucesso = 100% válida / 0% inválida.
CA002	A chamada automática deve ser liberada nos horários configurados.	App deve ativar botão de presença exatamente no horário previsto (~10 segundos de tolerância).
CA003	A resposta de presença deve ser registrada apenas se o aluno estiver no campus.	Teste com GPS → aceitação somente dentro de 100m de raio do local definido.
CA004	Relatórios CSV devem ser exportados corretamente.	Arquivo gerado deve conter data, horário, nome do aluno e status (Presente/Atrasado/Faltou).

CA005	O tempo de carregamento entre telas deve ser menor que 3 segundos.	Medição com cronômetro manual ou logs de performance Flutter.
CA006	O app deve permanecer funcional mesmo com perda temporária de internet.	Simulação de desconexão → dados devem ser sincronizados ao reconectar.
CA007	O design deve seguir o padrão visual do Figma fornecido.	Comparação entre protótipo e build final → tolerância visual ≤ 5% de diferença.